

## ネームサーバを用いた負荷分散方式

4 G-6

下川 俊彦\* 吉田 紀彦\*\* 牛島 和夫\*

\* 九州大学大学院システム情報科学研究科

\*\* 長崎大学工学部

## 1. はじめに

近年インターネットでは WWW を始めとする様々なサービスの提供が行われるようになってきた。これに伴い、ユーザ数も爆発的に増加している。この結果、一部のサーバに負荷が集中するようになってきている。

これに対処するため、サーバを複数化することによる負荷の分散が試みられている。サーバの複数化を行った場合、どのようにしてユーザを振り分けるかが問題となる。

既存の方法はユーザに過度の負担を強いたり、柔軟性に欠けるものである。そこで、本研究では柔軟性の高い負荷分散方式を開発する。既存のアプリケーションへの変更を最小限に押さえるために DNS を用いる。さらに proxy サーバ的に働く質問処理プロセッサを導入することで既存の DNS への変更を不要にする。

## 2. 複数化したサーバの選択方法

複数化したサーバの選択方法としては以下のようなものがある。

## 2.1 ラウンドロビン DNS による振り分け

DNS のラウンドロビン処理応答<sup>[1]</sup>では、単一のサービスホスト名に対して、複数の IP アドレスを割り当てた場合に、問い合わせ毎に異なる IP アドレスを答える。これを利用し、要求毎に異なるサーバに振り分ける。

しかし、この方法では単純なラウンドロビンによる割り当てしか行えない。このため、サーバの処理能力やクライアント・サーバ間のネットワークの状況（例えばネットワークポロジやネットワークのトラフィック量）に応じた振り分けを行うことはできない。

この方法の改善としては<sup>[2]</sup>がある。サーバ毎に処理能力に応じた数の IP アドレスを割り当てることで処理能力に応じた振り分けを行っている。しかし、この方法でもクライアント・サーバ間のネットワークの状況などに対応することはできない。

## 2.2 ユーザの選択による振り分け

複数化したサーバにそれぞれ異なる名前を付ける。例えばメインのサーバを www.provider.com とし、ミラー

サーバを mirror1.provider.com, mirror2.provider.com 等とする。そしてユーザにサーバの選択を任せる。

この方法は、ユーザに振り分けを委ねているため適切な負荷分散を行うことが困難である。また、ユーザは本来不必要な判断を強いられる。実装は簡単であるが、サーバの振り分け方法としては適さないと考える。

## 2.3 Proxy でのリクエスト書き換えによる振り分け

WWW の proxy サーバの一部は、要求されたリクエスト (URL) の一部を書き換えた後、実際の要求を行う機能を持つ。このような機能を用いてサーバの振り分けを行うことができる。

リクエストの書き換えアルゴリズムによって、様々な状況に対応することも可能である。しかし、この方法は利用するアプリケーションが proxy に対応している必要がある。このため proxy に対応していないアプリケーションの場合、アプリケーションの変更が必要になる。

## 2.4 サーバのリダイレクトによる振り分け

一旦リクエストを受け付けたサーバ（以下、主たるサーバ）が、そのリクエストを他のサーバに転送する。

この方法では、主たるサーバの判断で様々な状況に対応することが可能となる。ただし、主たるサーバがボトルネックとなる可能性が高く、スケーラビリティに欠ける。

## 2.5 経路制御による振り分け

CISCO 社の Distributed Director<sup>[3]</sup> は、経路制御情報を用いることにより、リクエストを近隣のサーバに振り分ける機能を持っている。

長野オリンピックの www サーバでは、複数のサーバに単一 IP アドレスを割り当てた上で、経路制御技術を用いて近隣のサーバへの振り分けを行った<sup>[4]</sup>。

これらはベンダー独自の機能であったり、容易には用いることが難しいという欠点を持つ。

## 3. DNS による動的なサーバの振り分け

本研究では、動的にサーバを振り分けるための枠組みとして DNS に注目する。DNS を用いることにより、アプリケーションは特別な対応を必要としない。

## 3.1 質問処理プロセッサの分離

既存の DNS は、ドメインネーム空間を管理するデータベース部分と、クライアントからの要求に答える質問処理プロセッサが一体となっている（図 1）。クライアントからの DNS 問い合わせに従って DNS データベースを検索し、その結果を返す。

Load balancing mechanism using Domain Name Systems. Toshihiko Shimokawa\*, Norihiko Yoshida\*\* and Kazuo Ushijima\*

\*Graduate School of Information Science and Electrical Engineering, Kyushu University.

\*\*Department of Computer and Information Sciences, Nagasaki University.

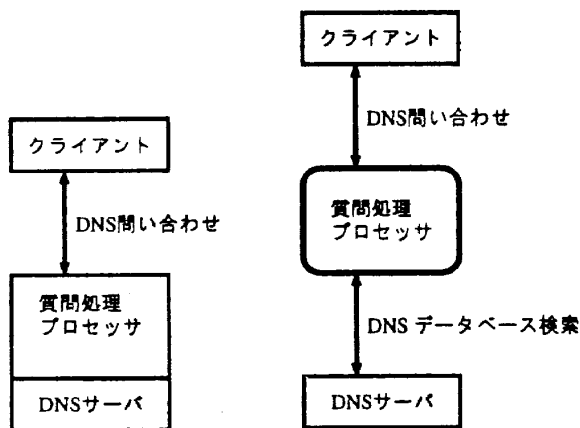


図 1: 従来のモデル

図 2: 本研究によるモデル

この両者を分離する(図2)。クライアントからのDNS問い合わせには答えるものの、単にDNSデータベースを検索するだけではなく、自律的な判断を行う。このことで動的なサーバの振り分けを行う。データベースの検索先として従来のDNSを利用する。このため、既存のDNSには手を加える必要がない。

質問処理プロセッサはDNSのproxyサーバと見ることもできる。クライアントは利用するDNSサーバを変更するだけで良い。DHCPやPPPを用いている場合、DNSサーバは自動で設定されることが多いため、この変更は全く意識しないが良い場合も多い。

### 3.2 多様なサーバ振り分け方式

現在設計・実装中の振り分けアルゴリズムは以下のような判断基準でサーバの選択を行う。

- i) ラウンドトリップタイムが一番短いもの
- ii) 経路情報的に一番近いもの
- iii) 負荷がもっとも低いもの
- iv) 管理者が最適と考えるもの

振り分けを行う候補としては、2.1. で述べた単一サービス名に付いた複数のIPアドレス群や、2.2. で述べた複数のミラーサーバ群を用いる。

## 4. 実装と評価

プロトタイプとしてオブジェクト指向スクリプト言語Ruby<sup>[5]</sup>を用いて実装を行っている。現在の実装で、全く判断を行わない(回答サーバは全てのリクエストをDNSサーバに転送する)場合の処理時間は、20ミリ秒未満である(Pentium II 266MHz, 64MB RAM, FreeBSD 2.2.6-RELEASE)。これが回答サーバを介すことによるオーバーヘッドである。DNSの問い合わせ時間全体や、その後の実際のリクエストの処理時間を考えると、これは無視できると考える。

管理者による選択を用いた場合の処理時間は、1ミリ秒未満である。これからも、回答サーバを導入することによるオーバーヘッドは無視して良いと考えることができ

る。

ユーザおよびサービス提供者から見た場合には、処理が最適なサーバに振り分けられるという利点がある。

## 5. おわりに

本研究ではDNSを用いて、複数化したサーバへの最適な処理の振り分けを行うシステムについて述べた。

今後の課題としては、以下の点がある。

- i) 様々なサーバ選択アルゴリズムの設計・実装
- ii) サーバ選択アルゴリズムの選択方法の検討
- iii) 選択候補の選択アルゴリズムの検討
- iv) さらに詳細なオーバーヘッドの計測
- v) 回答サーバ間の協調処理の検討

i) に関して、複数化したサーバの負荷情報をDNSに保持することも考えられる。ただし、その場合もDNSやアプリケーションへの改変を最小限に抑えるようにしたい。

iii) については、現在ほどのサーバがミラーサーバであるかを機械的に知る方法が存在しない。このため2.2のような方法でサーバの複数化が行われている場合に、回答サーバが単独で選択対象を決定することが不可能である。この解決策としては、ミラーサーバの情報をDNSに記述するなどの方法があるが、その記述方法を含め今後の課題である。

iv) については、現在のオーバーヘッドが小さい理由として、実装済のアルゴリズムが少ないことを挙げるができる。今後多様なアルゴリズムの実装を進めていった場合の処理速度については注意を払う必要がある。

v) については、ある程度大きな組織になると回答サーバが複数になると考えられる。その様な場合、各々の回答サーバがそれぞれ独自で判断をするのは必ずしも最適ではないと考えられる。このため回答サーバ間での協調処理が必要になる。今後はその方法も検討していく必要がある。

## 参考文献

- [1] T. Brisco, *DNS Support for Load Balancing*, RFC 1794, April 1995.
- [2] 馬場、山口: "DNSを用いた広域負荷分散の実装," 情報処理学会 分散システム運用技術研究会 研究報告 No.9, 1998
- [3] CISCO Systems Inc. *DistributedDirector*, 1997. Hypertext document. Available electronically at [http://www.cisco.com/warp/public/751/distdir/dd\\_wp.htm](http://www.cisco.com/warp/public/751/distdir/dd_wp.htm)
- [4] 重近、中村、笹川、村井: "長野オリンピックのネットワークと情報提供システム," 情報処理 Vol.39 No.2, 1998
- [5] Y. Matsumoto, *Ruby the Object-Oriented Script Language*, Hypertext document. Available electronically at <http://www.netlab.co.jp/ruby/>